Group 10

TRƯỜNG ĐẠI HỌC KHOA HỌC VÀ CÔNG NGHỆ HÀ NỘI

HTTP over RPC

(Act as HTTP proxy)

Members:

- **Đỗ Thanh Hiếu - BI10-059**
- **Trần Trung Hiếu - BI10-058**
- **Nguyễn Viết Nhân - BI10-132**
- **Đỗ Đình Phúc - BI10-137**
- **Nguyễn Tấn Dũng - BI10-041**

Table Of Contents

I. Introduction

- Overview

- HTTP over RPC (HTTP over Remote Procedure Call) is a protocol that allows a client on the Internet to connect securely to a Microsoft Exchange Server without having to log into a virtual private network (VPN) first.

- HTTP-RPC services are accessed by applying an HTTP verb such as GET or POST to a target resource. The target is specified by a path representing the name of the resource and is generally expressed as a noun such as /calendar or /contacts.

- Arguments are provided either via the query string or in the request body, like an HTML form. Results are generally returned as JSON, although operations that do not return a value are also supported.

- How it works?

- HTTP over RPC uses Secure Socket Layer (SSL) as a transport protocol and mandates that the server authenticates itself to the client using a digital certificate and associated private key. The protocol takes advantage of the Hypertext Transfer Protocol

(HTTP) connectivity built into the Outlook Web Access (OWA), foregoing requirements for an administrator to open additional ports on the corporate firewall.

- What is it used for?

- It allows developers to create and access HTTP-based web services using a convenient, RPC-like metaphor while preserving fundamental REST principles such as statelessness and uniform resource access.

II. Analysis & Design

1. HTTPS

- HTTPS is the abbreviation for hypertext transfer protocol secure, or secure hypertext transfer protocol.

- Unlike HTTP, HTTPS uses a secure certificate from a third-party vendor to secure a connection and verify that the site is legitimate. This secure certificate is known as an SSL Certificate (or "cert").

- SSL is an abbreviation for "secure sockets layer". This is what creates a secure, encrypted connection between a browser and a server, which protects the layer of communication between the two.

- This certificate encrypts a connection with a level of protection that is designated at your time of the purchase of an SSL certificate.

- An SSL certificate provides an extra layer of security for sensitive data that you do not want third-party attackers to access. This additional security can be extremely important when it comes to running e-commerce websites.

- It is used for secure communication over computer networks and is widely used on the Internet.

Figure 1.HTTPS and HTTP

2. HTTP Proxy

- An HTTP proxy server sits between a Web server (HTTP server) and the Web client.

- The HTTP proxy can essentially be described as a high-performance content filter that traffic flows through to reach you. In other words, it acts as an intermediary between the client browser and the destination web server. Then, any traffic that is processed through the server will appear as though it came from the proxy's dedicated IP address instead of the one that your device is associated with.

- It is used to process the HTTP protocol and checks for any potentially harmful content, before sending it to the Web client.

- HTTP proxies can be used to restrict the access to specific Internet resources, to bypass restrictions, or to hide your public IP address and achieve online anonymity.

- Furthermore, the HTTP proxy allows for a large number of users to utilize the connection at any one time, which makes it useful for companies that have a large number of employees. As a company, you can also add a layer of security by setting up an HTTP proxy server on your organization's public web server to stop attempts of storing unauthorized files.

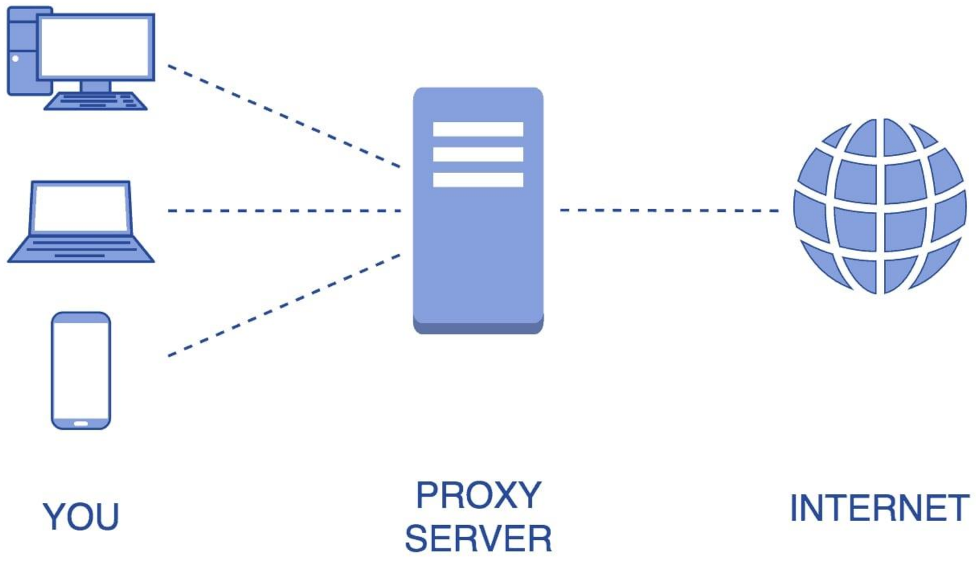


Figure 2 HTTP Proxy

3. RPC

- A remote procedure call is an interprocess communication technique that is used for client-server based applications. It is also known as a subroutine call or a function call.

- A client has a request message that the RPC translates and sends to the server. This request may be a procedure or a function call to a remote server. When the server receives the request, it sends the required response back to the client. The client is blocked while the server is processing the call and only resumed execution after the server is finished.

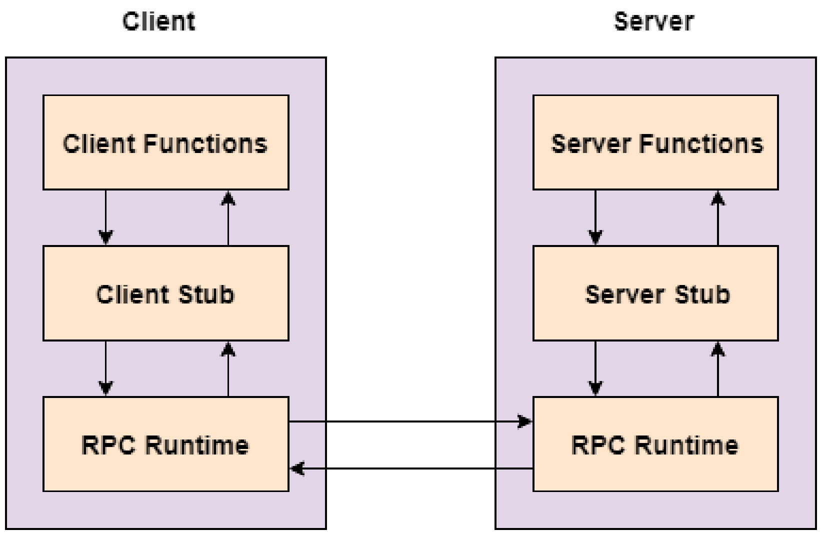


Figure 3 Remote Procedure Call (RPC)

- The sequence of events in a remote procedure call are given as follows:
    - The client stub is called by the client.
    - The client stub makes a system call to send the message to the server and puts the parameters in the message.

- The message is sent from the client to the server by the client's operating system.
- The message is passed to the server stub by the server operating system.
- The parameters are removed from the message by the server stub.
- Then, the server procedure is called by the server stub.

- RPCGEN is an interface generator pre-compiler for Sun Microsystems ONC RPC. It uses an interface definition file to create client and server stubs in C.

- RPCGEN creates stubs based on information contained within an IDL file. This file is written in a language called RPCL - remote procedure call language. This language closely mimics C in style, and is designed purely for defining specification to be used for ONC RPC.

- An RPC specification contains a number of definitions. These definitions are used by RPCGEN to create a header file for use by both the client and server, and client and server stubs.

4. Libcurl

- What is cURL: cURL, which stands for client URL, is a command line tool that developers use to transfer data to and from a server.
- At the most fundamental, cURL lets you talk to a server by specifying the location (in the form of a URL) and the data you want to send.

- Libcurl offers an API for parsing, updating and generating URLs. Using this, applications can take advantage of using libcurl's URL parser for its own purposes. By using the same parser, security problems due to different interpretations can be avoided.

- Libcurl offers an API for parsing, updating and generating URLs. Using this, applications can take advantage of using libcurl's URL parser for its own purposes. By using the same parser, security problems due to different interpretations can be avoided.

- We use Libcurl for HTTP request and response information.

III. Implementation

1. **How does an HTTP Proxy work:**
    - The browser (HTTP client) sends web requests to a proxy which in turn forwards it to the actual web server (HTTP server).

    - The web server sees the proxy server simply as another connection and answers it as usual.

    - The proxy server then delivers the HTTP response to the client. The communication uses standard ports like 80, 8080, or 3128.

    - The HTTP proxy also examines the source of the web traffic before sending it to an internal web client. Doing so ensures that potentially harmful content is far less likely to enter your network, and buffer overflow attacks can be avoided.

    - You can customize the HTTP proxy server's ruleset to suit your business requirements. Depending on the configuration, companies can set up the ruleset for different purposes, which will soon be discussed.

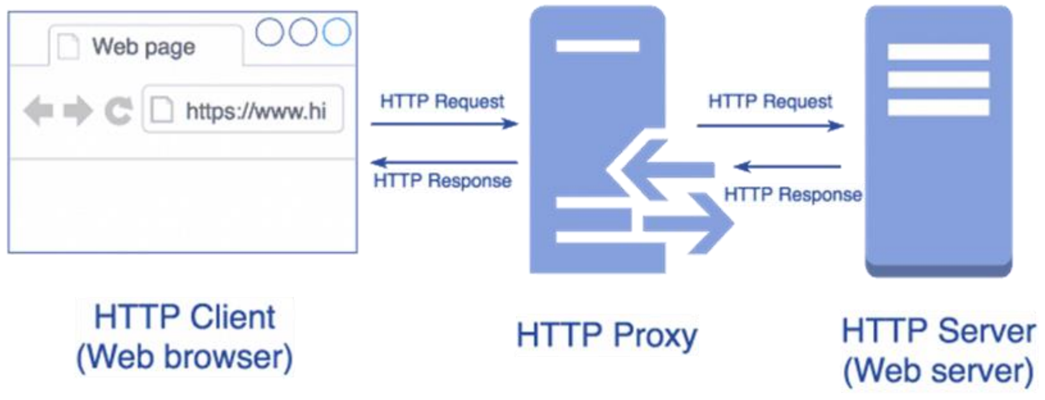


Figure 4 How a HTTP Proxy works (Client-proxy-server communication)

2. HTTP over RPC (act as HTTP proxy)

- The HTTP over RPC (act as HTTP Proxy) runs on an IIS (Internet Information Services) computer.

- It accepts RPC requests coming from the Internet, performs authentication, validation, and access checks on those requests, and if the request passes all tests, HTTP over RPC forwards the request to the RPC server that performs the actual processing.

- With HTTP over RPC the RPC client and server do not communicate directly; rather, they use the RPC Proxy as intermediary.

3. Our project process:

- Software used to implement the project:
    - Ubuntu with VMware.
    - C programming language with GNU C Compiler.
    - RPCGEN to create Remote Procedure Call.

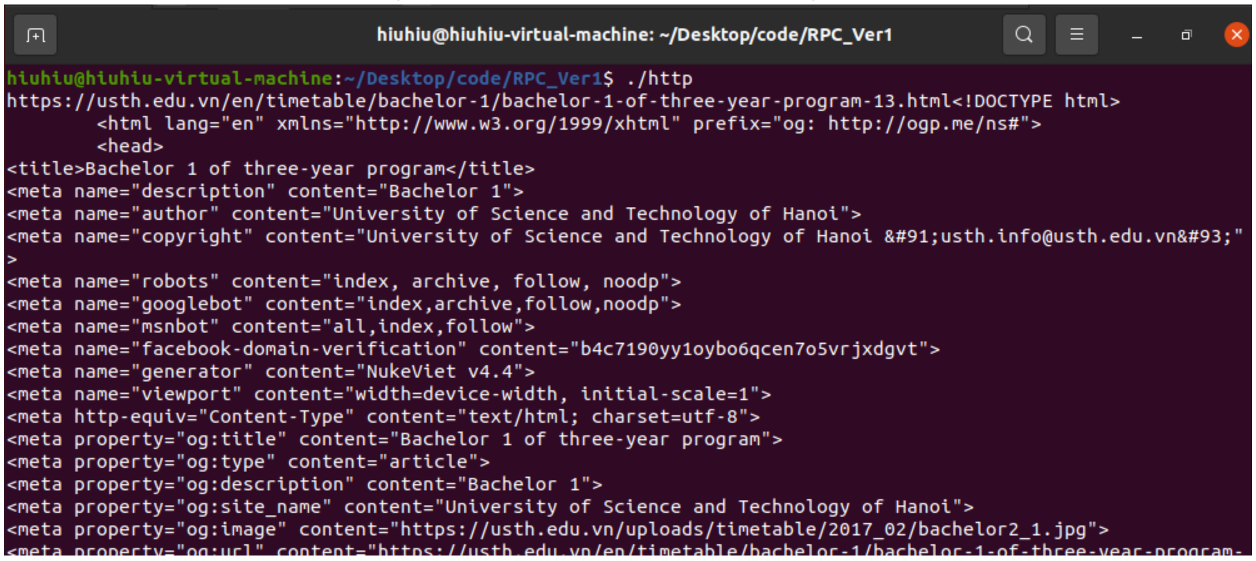
- Create method http request and response directly using Libcurl library.
    - Declare curl
    - Create curl
    - set options for a curl easy handle to request html doctype by url of this website.
    - Perform curl to get response (html doctype)
    - Print response result.

- Create RPC connection between client and server using RPCGEN.
    - Define program, struct, version of RPC.
    - Create file by using RPCGEN.
    - Check file server and client:
        - Check connection.
        - Print notification of status (connected or not).

- Combine the HTTP request and response method above into the generated RPC:
    - The client will pass the server address and the "url" of the web page to get information about.
    - The server will receive the "url" from the client and execute the HTTP request and response method and return the response result to the client.

IV. Result

- HTTP Request and direct response succeed.

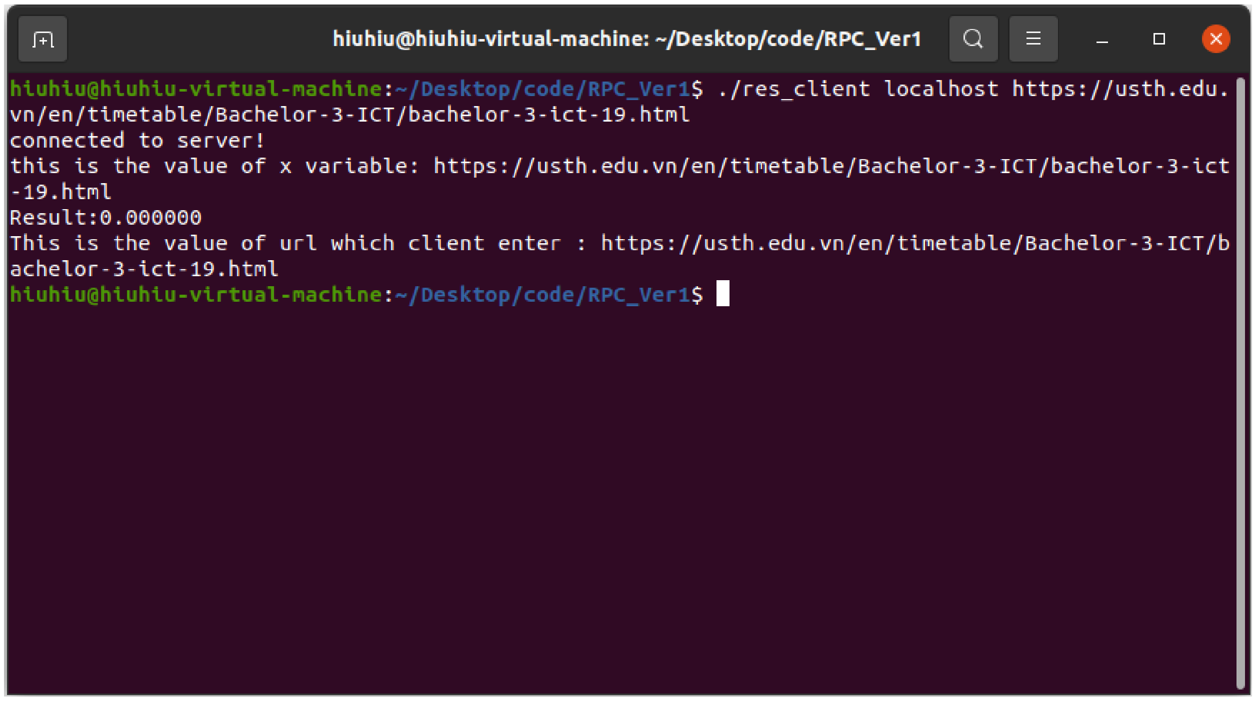


```
hiuhiu@hiuhiu-virtual-machine: ~/Desktop/code/RPC_Ver1
hiuhiu@hiuhiu-virtual-machine:~/Desktop/code/RPC_Ver1$ ./http
https://usth.edu.vn/en/timetable/bachelor-1/bachelor-1-of-three-year-program-13.html<!DOCTYPE html>
        <html lang="en" xmlns="http://www.w3.org/1999/xhtml" prefix="og: http://ogp.me/ns#">
        <head>
<title>Bachelor 1 of three-year program</title>
<meta name="description" content="Bachelor 1">
<meta name="author" content="University of Science and Technology of Hanoi">
<meta name="copyright" content="University of Science and Technology of Hanoi &#91;usth.info@usth.edu.vn&#93;"
>
<meta name="robots" content="index, archive, follow, noodp">
<meta name="googlebot" content="index,archive,follow,noodp">
<meta name="msnbot" content="all,index,follow">
<meta name="facebook-domain-verification" content="b4c7190yy1oybo6qcen7o5vrjxdgvt">
<meta name="generator" content="NukeViet v4.4">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta property="og:title" content="Bachelor 1 of three-year program">
<meta property="og:type" content="article">
<meta property="og:description" content="Bachelor 1">
<meta property="og:site_name" content="University of Science and Technology of Hanoi">
<meta property="og:image" content="https://usth.edu.vn/uploads/timetable/2017_02/bachelor2_1.jpg">
<meta property="og:url" content="https://usth.edu.vn/en/timetable/bachelor-1/bachelor-1-of-three-year-program-
```

Figure 5 : HTTP Request and direct response

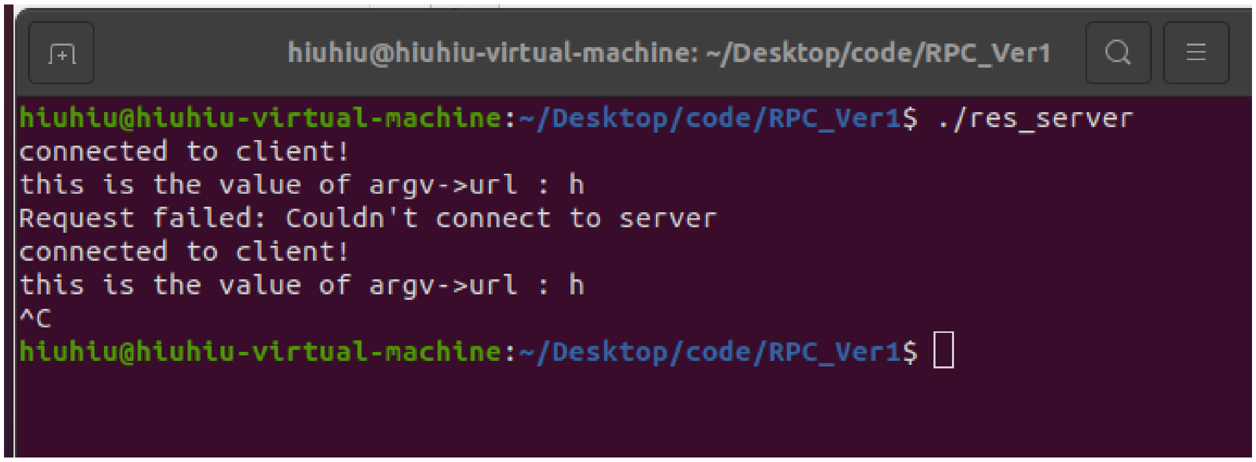
- In RPC, Server and Client have successfully connected to each other.
- HTTP over RPC not yet, the problem is:

- When we passing url from client to server, server just get only first letter of that url.
- So the url not correct make method http can not get url and response.



```
hiuhiu@hiuhiu-virtual-machine:~/Desktop/code/RPC_Ver1$ ./res_client localhost https://usth.edu.
vn/en/timetable/Bachelor-3-ICT/bachelor-3-ict-19.html
connected to server!
this is the value of x variable: https://usth.edu.vn/en/timetable/Bachelor-3-ICT/bachelor-3-ict
-19.html
Result:0.000000
This is the value of url which client enter : https://usth.edu.vn/en/timetable/Bachelor-3-ICT/b
achelor-3-ict-19.html
hiuhiu@hiuhiu-virtual-machine:~/Desktop/code/RPC_Ver1$
```

Figure 6: Client terminal (RPC and HTTP over RPC)



```
hiuhiu@hiuhiu-virtual-machine:~/Desktop/code/RPC_Ver1$ ./res_server
connected to client!
this is the value of argv->url : h
Request failed: Couldn't connect to server
connected to client!
this is the value of argv->url : h
^C
hiuhiu@hiuhiu-virtual-machine:~/Desktop/code/RPC_Ver1$
```

Figure 7: Server terminal (RPC and HTTP over RPC)

V. Conclusion & Future Works

- The project not done:
    - Complete HTTP over RPC.
        - Idea: using malloc() to define pointer contain url to pass from client to server.
    - System upgrade:
        - Handling exception.
        - Extend connections to multiple servers and clients.
        - Create graphical user interfaces for servers and clients.

VI. Github:

- *https://github.com/thanhhieudo278/DistributedSystems/tree/main/RPC_Ver1*

VII. Reference:

- *https://www.techtarget.com/searchapparchitecture/definition/Remote-Procedure-Call-RPC*
- *https://everything.curl.dev/libcurl-http*
- *https://www.youtube.com/watch?v=HbBxO5RXNhU*
- *https://oxylabs.io/blog/what-is-http-proxy?fbclid=IwAR34v0SRqac3AM0NdTSLsPm2iDTP_EUGjtGsyi3MZb1DB0UUjSU6KcohnzE*
- *https://docs.microsoft.com/en-us/windows/win32/rpc/using-http-as-an-rpc-*

transport?fbclid=IwAR2EFMSTwZBCnKTUoAtkgnHcpHu1XcNlK2-oQH3lBLUrTqUKq245yETgIh8

- https://www.tutorialspoint.com/remote-procedure-call-rpc?fbclid=IwAR3btx8pfYgMuVZBUNYLDoOUX4vOXxHkaeOBG2_1HepjI3fy5SnZVioUBXA
- https://hide-ip-proxy.com/what-is-an-http-proxy/?fbclid=IwAR1IJ9Y4ncdlyCo-xHCRQPgn8P0RWw_g98JKmBpRnpDg8EU3TNcGH1v2uYo#what-is-an-http-proxy